DOCUMENT RESUME

ED 183 183 IR 008 125

AUTHOR	Alessi, Stephen M.; Dennis, J. Richard
TITLE	Designing Instruction for Teaching with a Computer. The Illinois Series on Educational Application of Computers, No. 3e.
INSTITUTION	Illinois Univ., Urbana. Dept. of Secondary Education.
SPONS AGENCY	EXXON Education Foundation, New York, N.Y.
PUB DATE	79
NOTE	28p.; For related documents, see IR 008 123-139.
EDRS PRICE	MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS	*Computer Assisted Instruction; *Flow Charts; *Guidelines; *Instructional Design; Learning Activities; *Lesson Plans; Programing Languages; *Teacher Developed Materials; Teaching Methods
IDENTIFIERS	BASIC Programing Language; Computer Software

ABSTRACT

A review of the major components of lesson planning and two conventional planning methods provides background for this explanation of the type of planning required when designing instruction for teaching with the computer. This "formal planning" includes the identification and articulation of the logical sequencing of both instructional events and instructional decisions, and planning sequences are nonlinear, taking a flowchart form, rather than being tightly tied to the days of the week. Discussion of the factors involved in formal planning includes decisions as instructional events, sequencing based on decisions, the organization of formal planning, and constructing the flowchart; examples of first and second drafts of a flowchart for a series of lessons teaching the computer language BASIC are provided. This document was designed as a resource for preservice and inservice teacher training, and a study activity is provided, as well as a list of seven references. (CMV)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U S DEPARTMENT OF HEALTH, EDUCATION & WELFARE
NATIONAL INSTITUTE OF EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

ED183183

Number 3e

The Illinois Series on Educational Application of Computers

DESIGNING INSTRUCTION FOR TEACHING WITH A COMPUTER

Stephen M. Alessi

J. Richard Dennis

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY

J. Richard Dennis

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

College of Education
Department of Secondary Education
University of Illinois
Urbana, Illinois

This work was supported by a grant from
The EXXON Education Foundation

Copyright, 1979, Department of Secondary Education
University of Illinois

IR008125

To the reader,

This is one of a series of documents prepared as resources for the preservice and inservice training of teachers. The general theme of all the items in this series is "teaching with or about computers". The booklets are the collective work of a variety of staff and students (both present and former) of the College of Education, University of Illinois at Urbana-Champaing. The project generously was supported between 1977 and 1979 by a grant from the EXXON Education Foundation. All responsibility for the content of these documents rests with the authors and does not reflect necessarily the views of the EXXON Foundation. It was my extreme pleasure to direct this effort throughout.

J.Richard Dennis, PhD
Department of Secondary Education
University of Illinois

PLANNING INSTRUCTION FOR COMPUTER ASSISTANCE

Lesson plans are the details of what a teacher intends to do in teaching that topic or lesson. For computerized lessons, the plan would be the details of what the computer would do. In order to understand the process of planning for instructional computing, we first summerize the activities engaged in when planning a conventional lesson.

Nearly every teaching methods textbook includes a discussion of planning for teaching. Although the terminology and number of distinct phases of planning may vary, the major components of lesson planning are similar to the following:

1) Analysis -- Preparing a synopsis of the subject matter to be taught, a consideration of the student population, and a statement of the goals and objectives of the lesson.

2) Selection -- Choosing the instructional technique, the instructional materials and (perhaps) the instructional medium.

3) Development -- Writing the plan and preparing the needed materials.

4) Evaluation -- for grading the students, for deciding next lessons, for rating instructional materials and teaching methods.

The analysis one does in lesson planning varies from author to author. Some authors suggest the preparation of an outline of the subject matter. Others suggest more detail in the form of a careful concept or task analysis. The analysis would be focused on the concepts for which the student must demonstrate competence and would include mention of the means for evaluating such a demonstration of competence. When considering the student audience, activities range from thinking about the relevant characteristics of the students to planning for a pretest of prerequisite learnings. Finally, one must consider the resources and materials available for the intended instruction. Time is among the most important of these resources. One frequently must establish priorities among various instructional events and allocate varying amounts of time accordingly.

During the selection phase of instructional planning, one of the most important "selections" is the instructional technique to be used to present or structure the lesson. Runion (1972), in analysing methods of initiating lessons in mathematics, has identified the following strategies:

1) Stating Goals -- lesson begins by clearly stating goals.

2) Outlining -- lesson begins with a written or oral outline of major points.

3) Using an Analogue -- lesson begins with an analogue or model from the experience of the students and related to the lesson content.

4) Using Historical Material -- lesson begins with related historical information.

5) Reviewing Subordinate Information -- lesson begins with a review of prerequisites.

6) Giving Reasons -- lesson begins with an attempt at justifying study of the topic.

7) Presenting a Problematic Situation -- lesson begins with posing of a problem which embodies (or contradicts) the concepts to be presented.

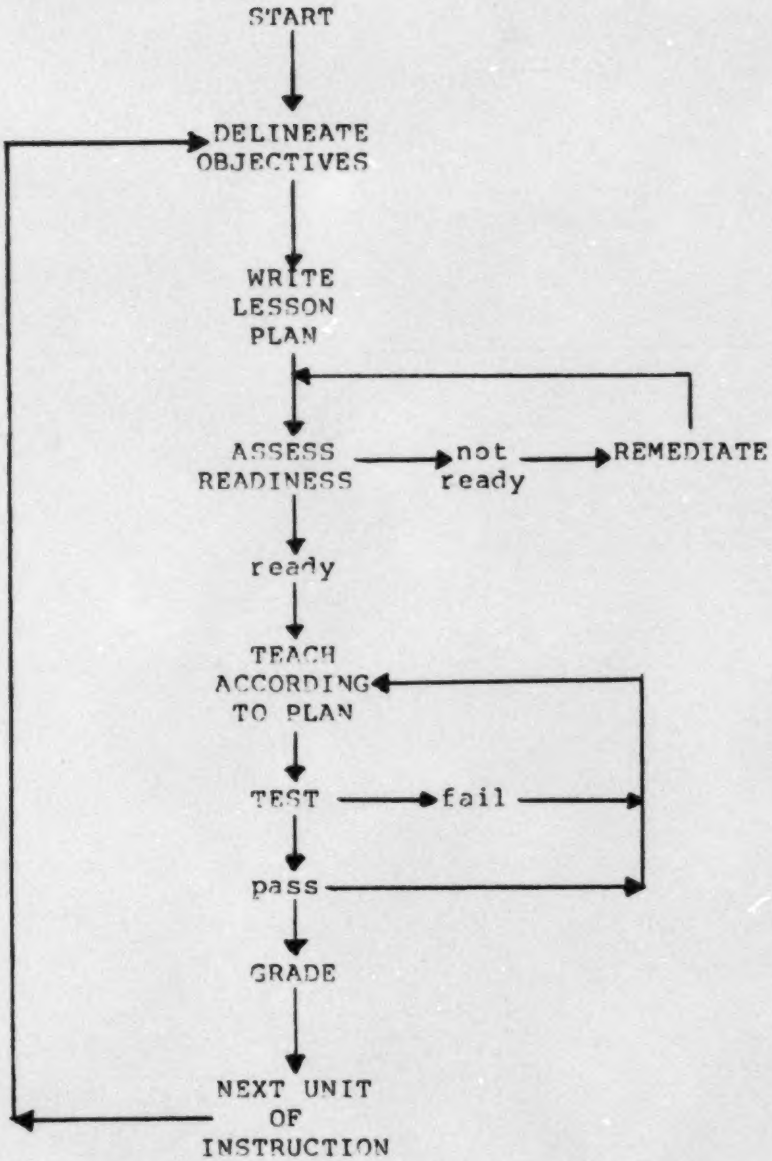
These items merely illustrate the variety of methods of opening a lesson. Subjects other than mathematics would produce similar lists of possibilities. Along with these options come the choices between the discovery and expository instructional methods, between individualized and group teaching, and among such other alternatives as play teaching, project or problem teaching (similar to number 7, above), programmed teaching, etc. Of course, all these choices must depend very heavily upon available materials and facilities.

The development phase of lesson planning involves the actual writing of the plan and preparation of needed resources. This stage varies greatly depending upon whether it is a lesson or an entire course that is being planned. For a lesson, one usually states the objectives in terms that will facilitate recognition of their achievement (sometimes called Behavioral Objectives). Following this are detailed descriptions of the lesson introductions, the examples (if any) to be used, the assignments and (perhaps) their follow-ups, and the conclusions of the lesson (whether by test, by interview, by homework evaluation, etc). Development of a course must go beyond these events to include sequencing of topics, establishing time lines, establishing a means of student accounting, etc.

Details of an evaluation plan must be separated depending upon the purpose of the evaluation. To evaluate students for grades, one must have a plan to observe or to test for acquired learnings. To evaluate for determining sequence of topics, one needs a means of assessing relative proficiencies on several preceding objectives. However, for each of these, one must decide what decision will be made, what data will be used as a basis, and how the data is to be gathered.

Methods of Planning a Lesson

Two general methods prevail for conventional lesson planning. First is a generalization of the procedures included in typical methods of teaching texts. This method assumes that a majority of the needed materials exist and are available to the teacher. The teacher begins with an analysis and listing of objectives (these may not be in behavioral terms). Based upon this analysis, the teacher selects some appropriate instructional materials and writes a lesson plan usually made up of a calendar that includes what materials are to be used on what days, what assignments are to be given, and what things are to be taught in class and read after class. Before instruction, the teacher takes some action to determine the students' readiness for the unit of instruction. Though this may be a pretest, it often is done using tests from a previous unit of instruction. Not much is said in the plan regarding students not deemed to be ready for the instruction, though presumably they are exposed to remedial instruction to raise their level of knowledge before continuing. Alternately, the teacher will divert from the plan during execution and spend a short time reviewing the prerequisite topics with the entire class. After the assessment of readiness, the teacher executes the lesson plan. At the end of the planned sequence of instruction, a test often is given. Based on the test (in part), the students are graded. Sometimes test performance is used to branch students to previous or new remedial instruction. Test results may recommend changing the choice of instructional materials for the next time students go through this sequence of instruction (such as next year). Students then go to the next instructional sequence. Figure 1 presents a flow diagram of this procedure.

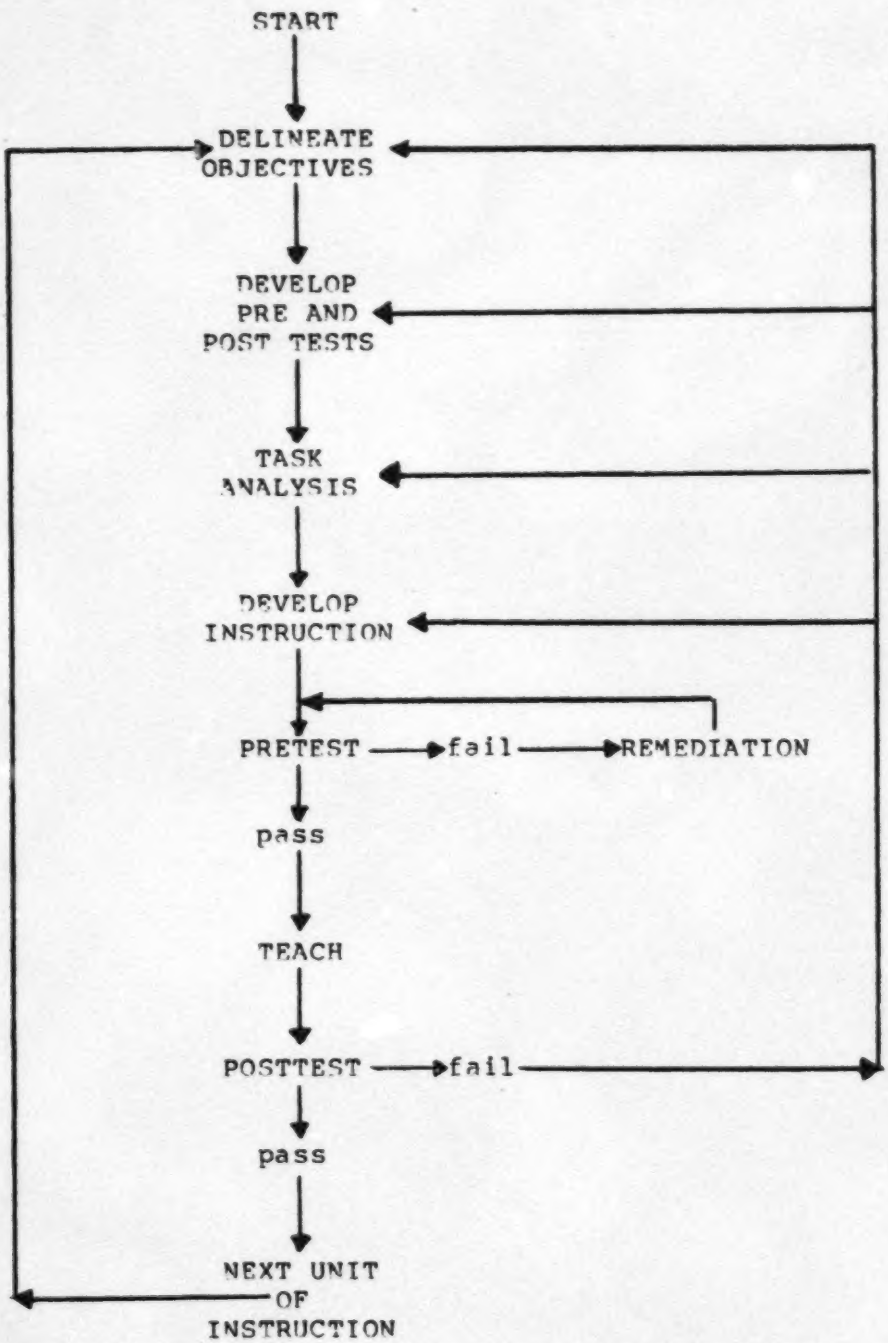


A CONVENTIONAL TEACHING ALGORITHM

FIGURE 1

The second popular instructional planning procedure is referred to as the educational technology method. This method is used mainly in situations in which instructional materials are being developed. In the educational technology approach, the developer begins with the delineation of behavioral objectives, including expected entering prerequisite behaviors, enabling objectives which are achieved in the course of instruction but are not the final goal, and terminal objectives which represent what the students should be able to do at the end of the particular sequence of instruction. The delineation of objectives leads directly to pretests and final tests. Such tests are recommmended to be criterion referenced through a statement of a standard for individual performance in contrast to norm referenced through a statement of the performance of a group. That is, the test should not be used to rank students in a class, but should say, for each individual, whether or not he has attained the objectives (or which objectives) of the instruction.

A task analysis is then undertaken. This represents an analysis of what skills must be taught, and in what order, towards eventually developing the skills represented by the terminal objectives. In essence, it represents a hierarchical analysis of skills which determines the teaching sequence. Instruction then is developed to teach each of the steps represented by the task analysis and objectives. At this point, the first phase of development ends and teaching begins. Students are pretested to determine if they meet the prerequisite (entering) behaviors. Students not meeting prerequisite behaviors are branched to remedial instruction (e.g. back to the previous unit of instuction or to some new instruction on the same materials). Students meeting prerequisite behaviors go into the sequence of instruction dictated by the task analysis. At the end of instruction, they are given the achievement test. Students passing this test go to the next unit of instruction, while students not passing are branched to remedial instruction. The achievement test data also are used to redevelop and improve the instructional materials. Sometimes this requires revision of the task analysis or list of objectives. Figure 2 presents a flow diagram of the educational technology model of instructional planning.



THE EDUCATIONAL TECHNOLOGY INSTRUCTIONAL ALGORITHM

FIGURE 2

Much of instructional planning follows one of these two models. In classrooms, these plans usually are descriptions (on the weekly or sometimes daily level) of things the teacher expects to do. Assignments that probably will be given and grading procedures that probably will be applied usually are included. Primarily, the sequence of instruction is based upon days or other measures of time by a calendar. Sequencing thus is fairly linear, proceeding from day (session) to day in a forward fashion. Typically, lesson plans for a course tend not to include branching based on student performance, details of what will be done with data from quiz results, scheduling considerations regarding the availablility of materials and facilities, etc. The result of such planning is that much that should be planned for is, in fact, not considered.

The reason for this lack of detail, of course, is that teachers are human beings, and properly trained and experienced human beings are very adept, on demand, at recognizing the need for a decision, at quickly assessing and weighing the alternatives, and at reaching an appropriate decision on an unplanned course of action.

Planning for Instructional Use of Computers

Computers are not human. Each action that a computer takes must have been anticipated, planned for, and relayed (through a program) to the computer by a human. The extent to which this program covers the range of imaginable events, conditions, and decisions determines the extent to which it will function properly when executed. This requires very careful planning.

In contrast to conventional planning, we wish to characterize what we call formal planning. The main characteristic of formal planning is the inclusion of attempts to identify and to articulate the logical sequencing of both instructional events and instructional decisions. This characteristic also is the major difference between formal and conventional style planning. A second difference is that while conventional planning sequences are centered around days of weeks, and are primarily linear, formal planning sequences are nonlinear, taking a "flowchart" form and are not tightly centered around days of the week.

Decisions as Instructional Events

As you read the following, you may wish to refer to the study activity on page 13 and relate this discussion to that activity.

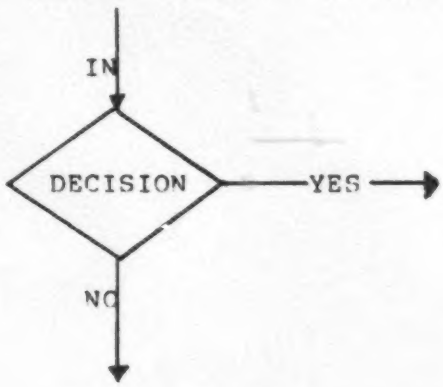
In the course of teaching, decisions constantly are made. When an instructor makes plans, without considering decisions, he may not take into consideration the consequences of the decisions on later aspects of instruction. For instance, when a lesson plan fills the week with the presentation of new material, but a quiz on Tuesday indicates that many students are not ready for the material on Wednesday, the instructor faces a decision for Wednesday of what to do with the students who are ready for the material and what to do with those who are not ready. The choice frequently is to present the new material to all students or to subject everyone to review (although some do not need it). In a more formal lesson plan, not only would the administration of the quiz be an event in the plan but also the decision regarding its outcome (pass or fail) is an event and alternative paths for different students are explicit parts of the plan. Thus building decision events into a teaching plan forces the instructor to consider all possible outcomes of each decision. Hopefully. taking decision points and branches into consideration forces the teacher to solve (and thus, to avoid) the Wednesday morning dilemma before Wednesday morning arrives. Avoiding such problems is not easy and often necessitates changes in the way the instructor usually conducts the class. The teacher who has used formal planning will realize that students doing poorly on the quiz had better be attended to sometime on Tuesday afternoon, or had better be assigned remedial homework. Facing the question of what to do when some students are not ready for Wednesday may also lead the teacher to use a more individually oriented approach to the course or, perhaps, decrease the amount of new material on Tuesday. The main point is that if everything is planned out before instruction begins, the instructor avoids periods in which activities are inappropriate for some students.

Sequencing Based on Decisions

Problems like the above are alleviated, in part, by the second aspect of formal planning. It encourages sequencing not on the basis of days, but on the basis of decisions. You don't go on to topic F because it is Thursday, but because your plan included a decision that it is time to go on to topic F. This is in direct analogy to the principle that a statement in a computer program is not executed because it is 10:15 PM but because the preceding statement said to branch to this one. Basing educational sequencing on decisions can lead to more effective instruction than basing sequencing entirely upon temporal conditions.

The Method of Formal Planning

A convenient (although by no means necessary) way of organizing formal planning is through the use of a flowchart. The flowchart is useful in representing a computer program because, in order for a computer program to work, all decisions that may be encountered must be planned for and the operations based upon the decision outcomes must be completely (formally) spelled out. A flowchart helps to do this. If a decision diamond only has a line heading out of the "yes" side, you know that the program will fail if that decision is ever answered "no", as it will be eventually. Otherwise, you would not have put a decision diamond there.



Similarly, if the instructional process comes up against a "no" decision when no procedures have been planned for a "no" decision, the instructional process will falter. Thus, just as a flowchart helps the programmer to produce a complete program which contains paths for all eventualities, the flowchart can help the instructional planner to produce a plan which accounts for all eventualities.

How is this done? It is not easy; in truth it never is done completely. The reason for this incompleteness is that it is generally impossible to predict all educational decisions and alternative events that will arise. The main rule, then, is to start with all the events and decisions that you DO expect and AT LEAST to include them in a flowchart.

The flowchart should be a standard flowchart in most respects. Of course, it should have a starting point. Any activity should be included and its temporal relation to other events taken into consideration in the flow of events. Decisions always should have paths leading out from all possible outcomes (including yes, don't know, pass, fail, marginal performance, student present, student absent, etc). The inclusion in the flowchart of all anticipated events and decisions is often a

tedious one. However, enough unanticipated events will occur to cause failure in the instructional process. It's a good idea to stack the cards as much in your favor as possible by not letting anticipated events cause problems.

What events are to be included? Class lectures and discussions, quizzes, use of quiz scores for decisions, decisions based upon class discussions, assignments given, correcting and grading of assignments, projects or student presentations, labs, exams, what to do when the student does not do an assignment, remedial activities, decisions when the student fails a second time, a third time, etc, the allocation of resources such as tests, library materials, computer terminals, instructor's office hours, and time for correcting papers and counseling students. The list goes on almost endlessly. The rule that everything anticipated should be included begins to sound a bit unrealistic. Indeed, it is an ideal. The more anticipated things you include, the better off you will be in that there are fewer chances for coming up against that Wednesday morning not knowing what to do.

There are several features to include in a flowchart for planning sequencing. The first, logically is to order events. Though this sounds obvious, it is easily violated. Be prepared to give quizzes on material taught, including the material covered by some student who has gone back along some remedial path and is behind the rest of the class. Plan for that contingency, even if it means administering quizzes on an individual basis. If that is so, you must take into consideration the logistics of the quizzing, the problem of cheating, etc.

Usually, subject matters contain a logical sequence. In learning the computer language BASIC, the syntax of assignment statements logically depends upon the syntax of arithmetic expressions, rules for variable names and constants, and the meaning of the symbol "=". Therefore, the teaching of these other topics should precede teaching assignment statements. Perhaps the decision based upon quizzes covering these prerequisite concepts also should precede teaching assignment statements.

The second consideration of sequencing (after logical ordering of events) is that of evaluation of the learner. With a very high achieving class you may be able to assume learning is taking place most of the time; usually you cannot. Quizzes and tests are always a good idea, and they should not be used solely for grading. Include evaluation mechanisms in the flowchart, along with sequencing paths for students passing and failing.

Collection of data (in contrast to the event which produces the data) is an often overlooked event. This is unfortunate because it is an important one. Data often is necessary for later decisions or for other aspects of the plan of instruction. If the

necessary data is not collected or cannot be retrieved, later decisions or events may be hampered. For instance, if you permit a student to take the final exam only after he has completed five quizzes, you must collect a record of the number of quizzes each student has completed. If the student must pass each of the five quizzes with a score of 80% or higher, then you also must record each quiz score and note whether the score reached the 80% level. If you wish to evaluate the difficulty of quizzes at the end of the course, you definitely must store the actual quiz scores somewhere. When your flowchart includes what you will do with data, you will be reminded to include what data to record, and thus will increase the likelihood of collecting the correct data.

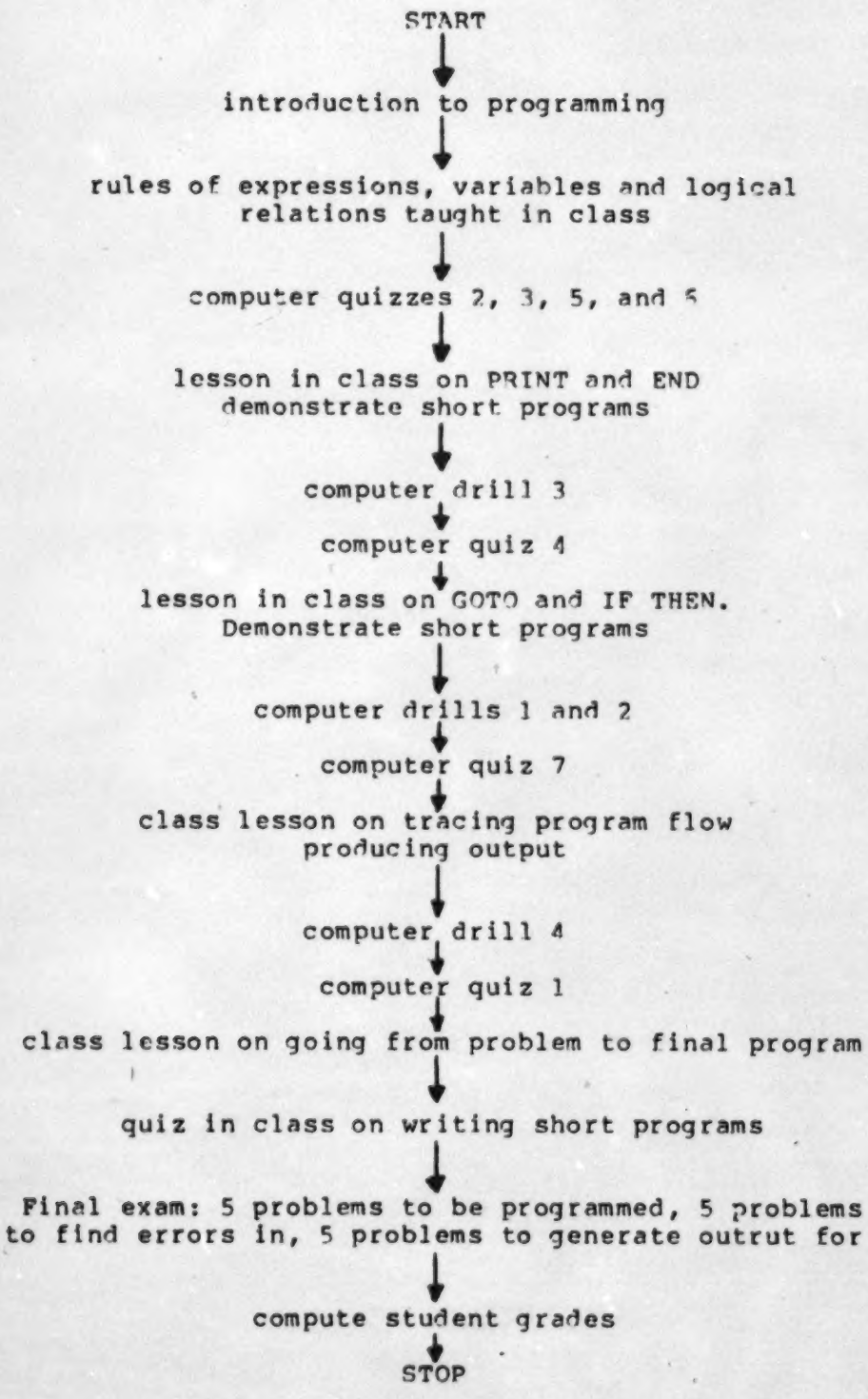
The end result is a standard looking flowchart. It will have a starting point, will have boxes describing the presentation of instruction, the giving of a quiz, the data collected on the quiz, the paths based on quiz performance, the remedial paths and the nonremedial paths, the other quizzes, tests, uses of quiz data, decisions regarding repeated failure, allocation of resources for students, etc. If computer facilities and techniques are included, there probably will be boxes teaching the student how to sign on to the computer and how to use it. There will be boxes for students using the computer for tutorials, drills, quizzes, tests, simulations, remediations, etc. There will be boxes about data collected on the computer or put into it. There will be decisions based on performance in computerized lessons. The more you include, the higher the likelihood that things will progress smoothly.

Constructing the Flowchart

Practically, you should not try to produce a complete flowchart on the first try. Many of the decisions that should occur early in instruction, and the data that should be collected early, will not be obvious until you recognize decisions that must be made later. Thus, your first draft might include mostly instructional events of the sort usually included in a lesson plan (presentation of material, quizzes, homework assignments, and grading procedures, for instance). A second draft will begin to include the less obvious events. Determining the grading procedure in the first draft will lead into determining what data to collect throughout the course from quizzes and tests. The order and nature of tests and quizzes will dictate necessary decisions about remediation in the second and third drafts. Data to be collected will lead to defining more precisely the nature of quizzes in the fourth draft. Regular instructional events will dictate a search by the instructor for appropriate instructional materials (if they are not already at hand) in the fifth draft. Events in each new draft will cause one to realize that other

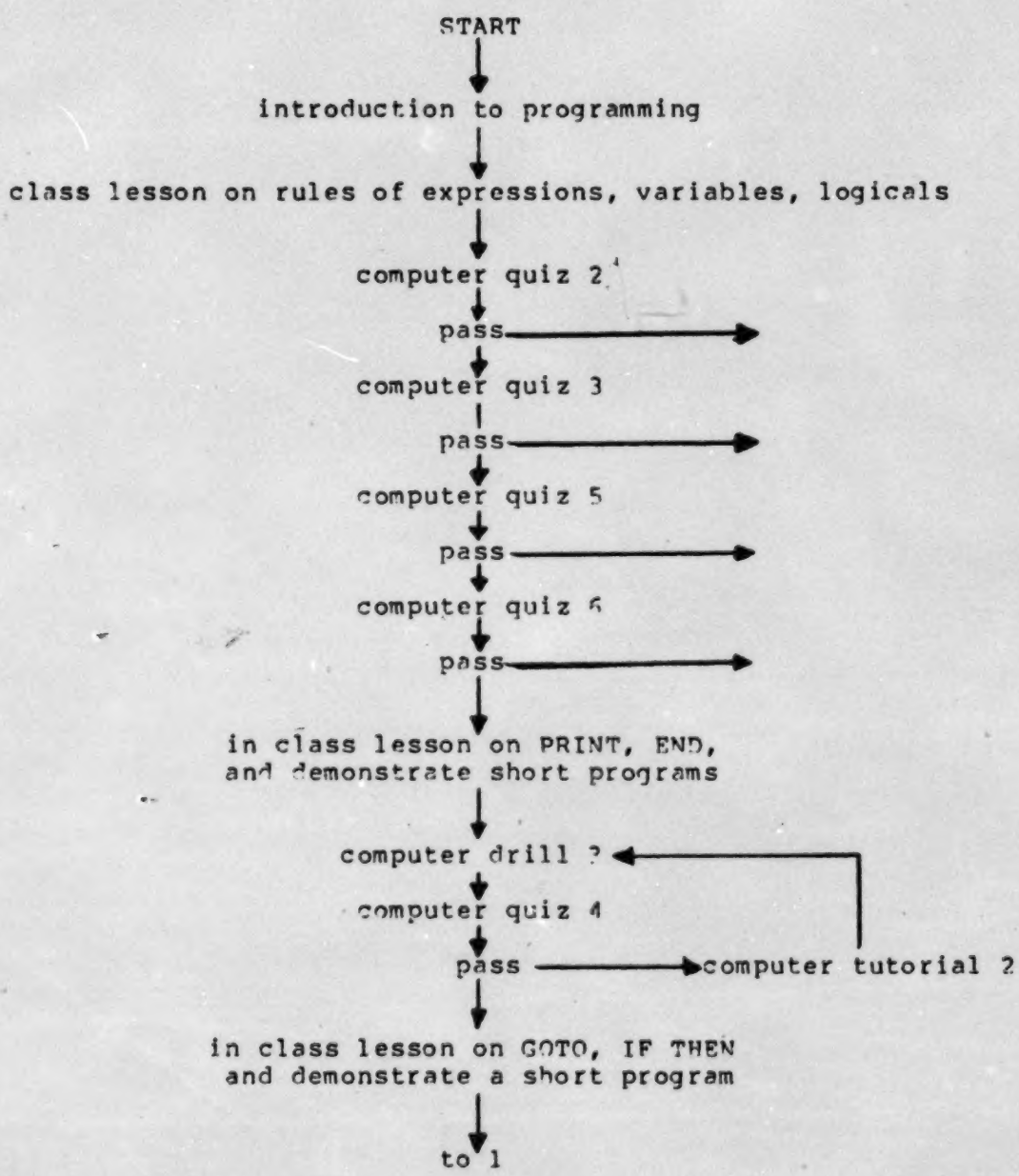
important events are missing, and later drafts will include them. At some point this will stop. You will feel that the flowchart accounts for a majority of the instructional contingencies (certainly a lot more than most informal lesson plans) and you are willing to risk the other unincluded contingencies.

Following are examples of first and second draft formal lesson plans (in flowchart form) for a series of lessons on teaching the computer language BASIC (you may wish to refer to the study activity while using the following flowcharts):


START
introduction to programming
rules of expressions, variables and logical relations taught in class
computer quizzes 2, 3, 5, and 6
lesson in class on PRINT and END
demonstrate short programs
computer drill 3
computer quiz 4
lesson in class on GOTO and IF THEN.
Demonstrate short programs
computer drills 1 and 2
computer quiz 7
class lesson on tracing program flow producing output
computer drill 4
computer quiz 1
class lesson on going from problem to final program
quiz in class on writing short programs
Final exam: 5 problems to be programmed, 5 problems to find errors in, 5 problems to generate outrut for
compute student grades
STOP

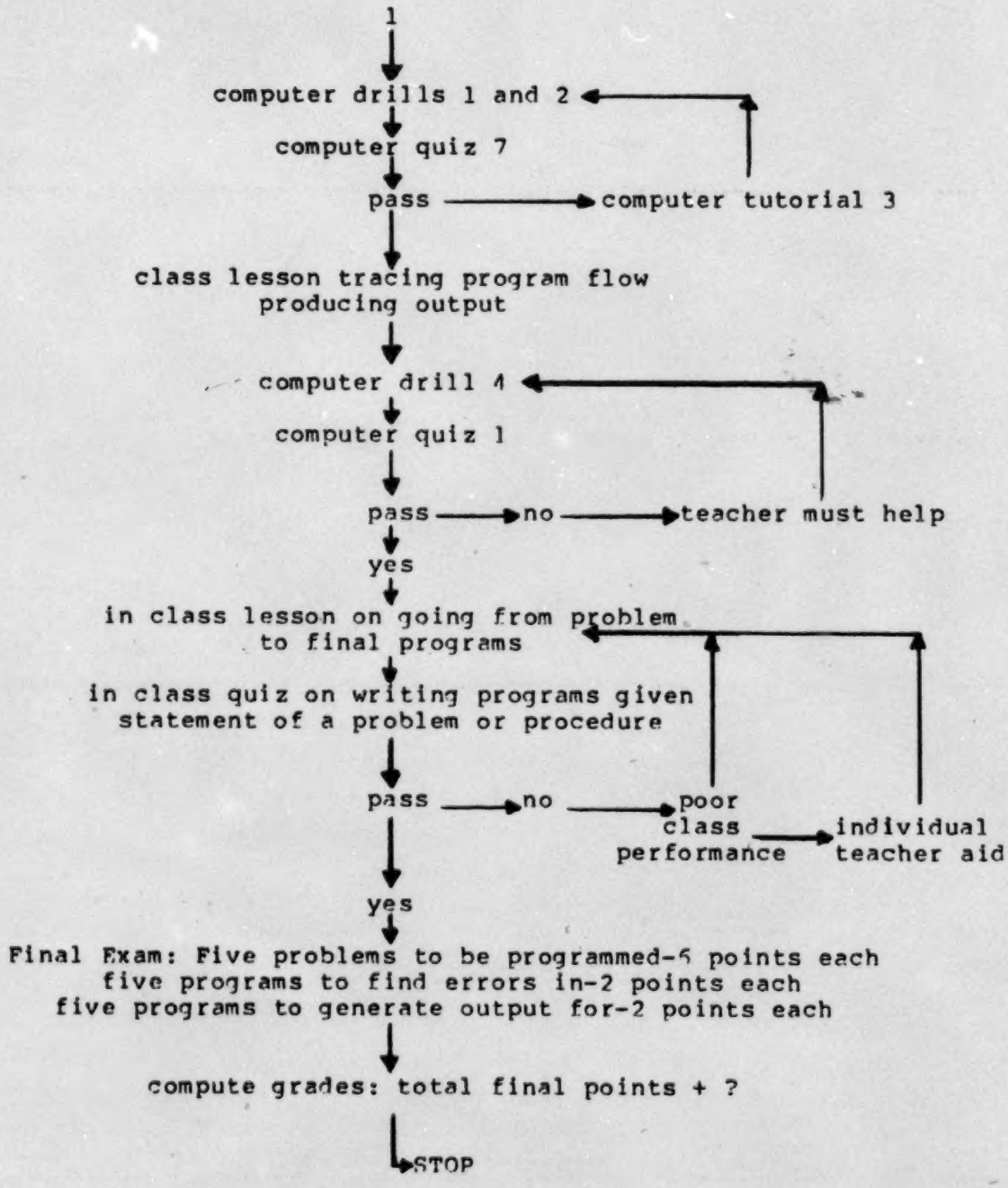
FIRST DRAFT FLOWCHART

This plan has several obvious omissions. For example, there are no pass-fail decisions after each qu.z. The scoring procedures for the final exam are missing. When these events are incorporated in the plan, our instructional flowchart looks like the following:



SECOND DRAFT FLOWCHART
(continued on next page)

NOTE: In making decisions on the four quizzes, it becomes apparent that they can be grouped and problems remediated for the group using computerized tutorials. More analysis is needed in future drafts on the storage of data and its use in selecting proper tutorial programs.


1
computer drills 1 and 2
computer quiz 7
pass
computer tutorial 3
class lesson tracing program flow
producing output
computer drill 4
computer quiz 1
pass
no
teacher must help
yes
in class lesson on going from problem
to final programs
in class quiz on writing programs given
statement of a problem or procedure
pass
no
poor
class
performance
individual
teacher aid
yes
Final Exam: Five problems to be programmed-6 points each
five programs to find errors in-2 points each
five programs to generate output for-2 points each
compute grades: total final points + ?
STOP

In this second draft, question marks point out questionable remediation decisions and paths. That these may be inappropriate is pointed out by the flowchart which helps us "see" things better than a conventional plan. The various paths we try help us to analyse the appropriateness of our remediation methods for the problem. Are we remediating the problems or just "doing something"? How do these path depend upon (or change due to) successive failures on the same material? Subsequent expansions of the flowchart might provide for such events as the allocation of terminal time to students, recording of quiz data for computing a final grade, improvements to the various remedial paths, evaluation procedures for computerizing material, etc.

Whether computerizing a single lesson activity or a larger portion of a course, this degree of detail to the plan is essential. To work effectively, an instructional computer must be informed of the events, the sequence, the decisions, and their consequences before instruction begins. Any of these overlooked can cause computer execution to halt unexpectedly, leaving the student at other than reasonable stopping points. Such omissions destroy student attitudes quickly and can aggravate greatly the management procedures of a classroom.

STUDY ACTIVITY

Prepare a lesson plan following the techniques of formal planning as best you can. Produce at least the first two flowchart drafts, plus a listing of all events, decisions, paths, problems, etc, which you can anticipate needing in later drafts. The following material describes the instructional setting.

You are the teacher of a class beginning their study of BASIC. The class meets five sessions per week for an hour each. Following is the subset of BASIC the students should learn:

Arithmetic operators

+ means 'positive' or 'add'
- means 'negative' or 'subtract'
* means 'multiply'
/ means 'divide'
** means 'to the power of ---'
() specify order of operations

Order of operations

1) inner parentheses first
2) ** is done next
3) * and / are done next (from left to right)
4) + and - are done last (from left to right)

Logical operators

'=' means 'is equal to'
'>' means 'is greater than'
'<' means 'is less than'
'>=' means 'is greater than or equal to'
'<=' means 'is less than or equal to'
'<>' means 'is not equal to'

Constants (numbers)

A constant must have from 1 to 9 digits.
If the constant is positive, it optionally may have a '+' to the left.
If negative, it MUST have a '-' to the left.
A decimal point is optional before the first, after the last or between any digits. Only one decimal point is permitted in a single constant.

Variable names

A variable name may be 1) any letter, or 2) any letter followed by any digit 0 through 9.

Form of lines in BASIC

Each line consists of a 1 to 5 digit number (the line number), followed by a blank, followed by a statement. A line must be 80 or fewer characters in length.

Assignment statements

Assignment statements evaluate an arithmetic expression (containing constants, variables and arithmetic operators) to a single constant, and put that result in a single variable.

Format: variable=arithmetic expression
Rules: Blanks may be used freely between operators, variables, and equals signs.
Only a single variable may appear on the left of the '=' sign.
Contiguous operators (other than parentheses) are not permitted except for + or - following *,/, or ** in which case + means 'positive' and - means 'negative'.
Contiguous variables and/or costants are not permitted.

Output statements

Format: PRINT (list of variables and literals separated by commas) Rules: Each print statement starts a new output line. A print statement without a list will just skip a line.
Blanks may appear between variables, commas, literals and the word PRINT.
A literal is a single quote followed by any string of symbols followed by a single quote. To put a single quote inside the literal, you must use two single quotes.
Literals will appear exactly as shown without the leading and trailing single quotes.
Variable names will be replaced by their numeric value at the time of the statement execution.
No spaces will come between members of the list on output unless put there using literals.
Decimal points will be put in numbers only if there are non-zero digits to the right of the decimal point.

Sequencing statement - GOTO

Format: GOTO (line number) or GO TO (line number)
Rules: The line number must belong to a line appearing in the program.
Any number of spaces may precede or follow the GO and the TO.
This statement will cause execution to transfer to the statement with the line number indicated.

Sequencing statements IF - THEN

Format: (arith exp)(logical rel) (arith exp) THEN (line number)
Rules: Any number of spaces may precede the IF. At least one space is required after it.
At least one space precedes and follows the THEN.
The line number must correspond to one of the statements in the program. The arithmetic expression may be permitted in an assignment statement.
Only one logical relation is permitted.
Usage: The two expressions are evaluated to single constants and the constants compared to see if the logical relationship is true or false. If false, the next statement in the program is executed. If true, execution transfers to the statement with the line number indicated after the THEN.

END statement

The END statements stops the execution of the program. It must be the last line in the program and have the largest line number.

Format : END
Rules: Any number of spaces are permitted before or after the END.

Caracter set

The following characters are permitted:
Letters: abcdefghijklmnopqrstuvwxyz
Digits: 0123456789
Special: + - * / = " , ? ; : ( ) space and period.

Execution order rules

Execution begins at the first statement and normally proceeds from each statement to the one immediately following (i.e. having the next largest line number).
GOTO or IF THEN statements may branch to a line other than the next one, after which execution continues normally.
Execution ends when the END command is encountered, or an error occurs. (errors include dividing by zero, overflow, and underflow)
Programs with syntax errors will not begin execution.

The general goal of the course is that the students be able to code and debug simple BASIC programs on paper. The students have no computer science background and have had mathematics through high school algebra. The course is offered through the mathematics department, which has a schoolwide rule that all mathematics courses grade according to the following formula: 50% of the grade is the final exam, and the rest of the grade is five quizzes worth a total of 50% (but not necessarily 10% each). As this course is an elective, your principal feels that despite the objective grading formula, all students are expected to get A, B, or C. A is defined as 90% to 100%, B as 80% to 89%, C as 70% to 79%, D as 60% to 69% and failing as below 60%.

Finally, the teachers in the math department are interested in the effectiveness of the computerized materials available. Though you may use any or none if you wish, it has been requested that whatever components are used be evaluated and data supplied to the department on the effectiveness. This data of course, should NOT be from computerized components themselves.

The teacher who previously taught the course gave you a brief description (perhaps not entirely accurate) of what was done before. This teacher admitted to having some problems and students did not do as well on the final exam as desired. It also was reported that students did well on the quizzes from week 2 and week 6, but poorly on the quizzes from week 3. Furthermore, students did well on the in-class part of the final but very poorly on the take-home part. The previous teacher's plan is as follows:

WEEK 1 Overview and organization - In class, explained what programming is all about, what computers are used for, overview BASIC and the content of the rest of the course.

WEEK 2 Rules concerning numbers and variables - In class, syntax rules were described and practiced and computerized Quizzes 2 and 3 were taken.

WEEK 3 Commands and their syntax, semantics - In class, the main commands (assignments, GOTO, IF THEN, PRINT, and END) were explained in terms of proper syntax and uses. Out of class, students were expected to use computerized Tutorials 1,2 and 3; Drill 1; and Quizzes 1 and 4.

WEEK 4 Flowcharting - The week primarily was spent on the theory of algorithms, problem solving, and flowcharting algorithms. Class time was used to teach the symbols and their purposes in flowcharting. Out of class, students were given a variety (about four) of problems to be flowcharted.

WEEK 5 Programming, translating flowcharts into statements - Class time was used to show how the flowcharts produced in the previous week could be translated into program statements. As an out of class assignment, three new flowcharts were given which the students were to translate into programs.

WEEK 6 Debugging and Final Exam - The first three classes were used to generally describe tracing the flow of programs, inserting debug flag statements, etc. Quiz 1 was done during the third class. On Thursday the final was described and discussed. The final started on Friday with an in-class portion consisting of five programs in which the students were to find errors. A take-home portion to be handed in on Monday consisted of English descriptions of five problems or procedures (simple math, science, accounting, etc) to be programmed.

The following computerized programs were available to the previous teacher and are available to you. Your school has a small minicomputer with five teletype terminals. There will be about 10 to 15 students in your class. One set of programs is available which contains three tutorials (linear programmed instruction), four drills, seven quizzes, and one fairly long test. You can assign any or all of these fifteen options to each student individually, and may change the options as often as you wish. The student can choose to do only those options you assign him and can do each only once. Following are the details of the options, including average expected execution times:

Tutorial 1 Rules for variable names, constants, and arithmetic expressions. 30 minutes.

Tutorial 2 Syntax and meaning of assignment and output statements, with statement numbering and arithmetic operators. 30 minutes.

Tutorial 3 Syntax and meaning of branching statements and the END command. Assumes knowledge of statement numbering. Also teaches logical operators. 30 minutes.

Drill 1 Syntax of all statements. The student is shown statements and must choose one of five alternative answers. The four alternatives are the possible corrected forms, and one additional alternative saying that the original statement was correct. The students must determine which corrected form is in fact correct or say that the original is correct. There are 50 items. Students spend as much time as desired getting problems by random selection.

Drill 2 Producing statements. The student is given an English command and must choose the one of four BASIC commands which satisfies the English command or say that none of them satisfy it. There are 50 items, same format as Drill 1.

Drill 3 Generating output. The student is given a short program using only the assignment, PRINT, and END commands. Each program has four alternative outputs. The student must choose the correct output for the program or say that none are correct. There are 30 items, same format as Drill 1.

Drill 4 Generating output. The student is given a short program using all commands (those in Drill 3 plus the GOTO and IF THEN) and again must choose the correct of four possible outputs or say that none are correct. There are 30 items as in Drill 1.

Quiz 1 Generating output (same as Drill 4). The student must do 10 problems, 1 point each.

Quiz 2 Syntax. Each problem shows a combination of five variable names, constants, and arithmetic expressions. The student must identify the syntactically incorrect one. 10 problems, 1 point each.

Quiz 3 The student is given problems with arithmetic expressions containing only constants and must choose the correct numeric answer. 10 problems, 1 point each.

Quiz 4 Problems consist of five assignment and output statements. The student must identify the syntactically incorrect one. 10 problems, 1 point each.

Quiz 5 Each problem displays an arithmetic or logical operator. The student must choose the English phrase indicating its meaning. 20 problems, one half point each.

Quiz 6 Each problem displays an arithmetic expression. Alternatives are five English descriptions of the order of operations. Students are to choose the correct one. 5 problems, 2 points each.

Quiz 7 Each problem consists of five sequence commands (GOTO'S or IF THEN'S). The student must choose the syntactically incorrect one. 10 problems, 1 point each.

Test 25 multiple choice problems. 10 are of the Quiz 1 and Drill 4 variety. 15 consist of a program with five possible errors (including syntax, endless loops, no output, etc). The student must indicate the error. Each question is worth 2 points on the final score. (note: the previous teacher did NOT use this exam).

BIBLIOGRAPHY

Anderson, R.C. and Faust, G.W., Educational Psychology: The Science of Instruction and Learning, Dodd, Mead and Company, New York, 1973. More emphasis on instructional development in the educational technology vein. Part one is instructional techniques and development. Part two is on teaching and learning techniques.

Cooney, Davis and Henderson, Dynamics of Teaching Secondary School Mathematics, Houghton Mifflin Company, Boston, 1975. In addition to planning and methods unique to mathematics, discusses special techniques for fast and slow learners. Many other methods texts do this also.

Engelmann, Siegfried, Conceptual Learning, Adapt Press, Inc, Sioux Falls, South Dakota, 1969. A short book on instructional development.

Hook, J.N., The Teaching of High School English, The Ronald Press, New York, 1965. Fairly traditional planning and methods specific to English instruction.

Jefferies, D.J., Lesson Planning and Teaching, Home and School Press, Titusville, New Jersey, 1966. Describes planning methods, curriculum planning, methods of motivating students, use of A-V equipment for teaching in general and a little more on evaluation.

Kochhar, S.K., Methods and Techniques of Teaching, Sterling Publishers, New Delhi, India, 1967. Describes planning and methods, curriculum planning, methods of motivating students, use of A-V equipment for teaching in general.

Runion, Garth E., "The Development of Selected Initiating Activities in the Teaching of Mathematics", Dissertation for The University of Illinois, Urbana, Illinois, 1972, DA no 34A pp 653, August, 1973.